IN THE CLAIMS:

1-5. (canceled).

6. (currently amended) A method, for recognizing a pattern in a~~n~~ register transfer level (RTL) description of an integrated circuit (IC) design, comprising:

a) identifying a pattern correspondence element in a pattern instance;

b) building a pattern tree corresponding to the pattern instance;

c) defining a list of candidate design correspondence elements in a design instance of the IC, wherein rare design correspondence elements are chosen from the list for comparison at a ~~higher~~ highest priority; and

d) iteratively, for each ~~design correspondence~~ element in said list of candidate design correspondence elements, comparing each rank in a tree representation of said design instance built around said each design correspondence element with corresponding rank in said pattern tree;

wherein the comparing is performed using a sub-process comprising:

d-1) selecting a rank for each node from the tree representation of said design instance to be matched;

d-2) iteratively, for each node with the selected rank building a child node list;

d-3) sorting said child node list;

d-4) building a list of pattern nodes with the selected rank;

d-5) matching between said list of pattern nodes and said child node list; and,

d-6) repeating said steps d-1) through d-5) for all ranks in said design instance;

wherein said child node list comprises terminals of said design instance;

wherein, when comparing, nodes in said list of pattern nodes are matched against nodes in said child node list according to said nodes' order in said list of pattern nodes and said child node list;

wherein said nodes are considered matched if said nodes have the same identification name;

wherein said identification name comprises at least a terminal name; and

wherein a report is generated indicating matched nodes.

7-10. (canceled).

11. (previously presented) The method of claim 6, wherein the pattern tree is built using a sub-process comprising:

a) iteratively, for each node starting from said pattern correspondence element building a child node list;

b) for each node in said child node list, setting a rank level of said each node; and c) sorting said child node list.

12. (original) The method of claim 11, wherein a parent node is attached to said child node list of said parent node.

13. (original) The method of claim 12, wherein said child node list comprises a list of fanin terminals.

14. (original) The method of claim 11, wherein said fanin terminals comprise a list of input terminals.

15. (original) The method of claim 12, wherein said child node list comprises a list of fanout terminals.

16. (original) The method of claim 15, wherein said fanout terminals comprise a list of output terminals.

17. (original) The method of claim 11, wherein said rank level is a number of terminal steps from said pattern correspondence element to a node having the rank.

18. (original) The method of claim 17, wherein said terminals include at least one of fanin terminals and fanout terminals.

19. (original) The method of claim 11, wherein said rank level is set to a rank level of a parent node of said child node list incremented by one.

20. (original) The method of claim 19, wherein said rank level of the root of said pattern tree is set to zero.

21. (original) The method of claim 19, wherein said sorting comprises a lexical sorting.

22. (canceled).

23. (canceled).

24. (Currently Amended) The method of claim ~~22~~6, wherein a parent node is attached to said child node list of said parent node.

25. (Currently Amended) The method of claim ~~22~~6, wherein permutable terminals are assigned equivalent node names.

26. (original) The method of claim 25, wherein said child node list comprises a list of fanin terminals of said parent node.

27. (Currently Amended) The method of claim ~~22~~6, wherein said child node list comprises a list of fanout terminals of said parent node.

28. (Currently Amended) The method of claim ~~22~~6, wherein said sorting comprises a lexical sorting.

29-31. (canceled).

32. (Currently Amended) The method of claim ~~30~~6, wherein, when a node is not matched, a report ~~indicating failure~~ is generated indicating a failure to match~~of nodes are mismatched~~.

33-38. (canceled).

39. (currently amended) A computer program product including computer-readable media with instructions to enable a computer to implement a process, for recognizing a pattern in a register transfer level (RTL) description of an integrated circuit (IC) design, the process comprising:

a) identifying a pattern correspondence element in a pattern instance;

b) building a pattern tree corresponding to the pattern instance;

c) defining a list of candidate design correspondence elements in a design instance of the IC, wherein rare design correspondence elements are chosen from the list for comparison at a ~~higher~~ highers priority; and

d) iteratively, for each ~~design correspondence~~ element in said list of candidate design correspondence elements, comparing each rank in a tree representation of said design instance built around said each design correspondence element with corresponding rank in said pattern tree;

wherein the comparing is performed using a sub-process comprising:

d-1) selecting a rank for each node from the tree representation of said design instance to be matched;

d-2) iteratively, for each node with the selected rank building a child node list;

d-3) sorting said child node list;

d-4) building a list of pattern nodes with the selected rank;

d-5) matching between said list of pattern nodes and said child node list; and,

d-6) repeating said steps d-1) through d-5) for all ranks in said design instance;

wherein said child node list comprises terminals of said design instance;

wherein, when comparing, nodes in said list of pattern nodes are matched against nodes in said child node list according to said nodes' order in said list of pattern nodes and said child node list;

wherein said nodes are considered matched if said nodes have the same identification name;

wherein said identification name comprises at least a terminal name; and

wherein a report is generated indicating matched nodes.

40-43. (canceled).

44. (previously presented) The computer program product of claim 39, wherein the pattern tree is built using a sub-process comprising:

a) iteratively, for each node starting from said pattern correspondence element building a child node list;

b) for each node in said child node list, setting a rank level of said each node; and

c) sorting said child node list.

45. (original) The computer program product of claim 44, wherein a parent node is attached to said child node list of said parent node.

46. (original) The computer program product of claim 45, wherein said child node list comprises a list of fanin terminals.

47. (original) The computer program product of claim 44, wherein said fanin terminals comprise a list of input terminals.

48. (original) The computer program product of claim 45, wherein said child node list comprises a list of fanout terminals.

49. (original) The computer program product of claim 48, wherein said fanout terminals comprise a list of output terminals.

50. (original) The computer program product of claim 44, wherein said rank level is a number of terminal steps from said pattern correspondence element to a node having the rank.

51. (original) The computer program product of claim 50, wherein said terminals include at least one of fanin terminals and fanout terminals.

52. (original) The computer program product of claim 44, wherein said rank level is set to a rank level of a parent node of said child node list incremented by one.

53. (Currently Amended) The computer program product of claim ~~54~~ 44, wherein said rank level of the root of said pattern tree is set to zero.

54. (original) The computer program product of claim 52, wherein said sorting comprises a lexical sorting.

55. (Canceled).

56. (Canceled).

57. (Currently Amended) The computer program product of claim ~~55~~39, wherein a parent node is attached to said child node list of said parent node.

58. (Currently Amended) The computer program product of claim ~~55~~39, wherein permutable terminals are assigned equivalent node names.

59. (original) The computer program product of claim 58, wherein said child node list comprises a list of fanin terminals of said parent node.

60. (original) The computer program product of claim 59, wherein said child node list comprises a list of fanout terminals of said parent node.

61. (Currently Amended) The computer program product of claim ~~55~~39, wherein said sorting comprises a lexical sorting.

62-64. (Canceled).

65. (Currently Amended) The computer program product of claim ~~63~~39, wherein, when a node is not matched, a report ~~indicating failure~~ is generated ~~of nodes are mismatched~~indicating a failure to match.

66-71. (canceled).

72. (currently amended) A system, for recognizing a pattern in a register transfer level (RTL) description of an integrated circuit (IC) design, comprising:

a compiler adapted to generate a pattern instance and a design instance;

a correspondence element identifier adapted to identify correspondence ~~element~~ elements in the pattern instance and in the design instance, wherein rare design correspondence elements are chosen for a comparison at a ~~higher~~ highest priority;

a tree generator to generate a pattern tree and a tree ~~representing~~ representation of the design instance around the correspondence element; and

a comparison unit adapted to iteratively compare a rank in the tree representation of said design instance with a corresponding rank in said pattern tree;

wherein the comparing is performed by the comparison unit according to a sub-process comprising:

d-1) selecting a rank for each node in the tree representation the design instance;

d-2) iteratively, for each node with the selected rank, building a child node list;

d-3) sorting said child node list;

d-4) building a list of pattern nodes with the selected rank;

d-5) matching between said list of pattern nodes and said child node list; and,

d-6) repeating said steps d-1) through d-5) for all ranks in said design instance;

wherein said child node list comprises terminals of said design instance;

wherein, when comparing, nodes in said list of pattern nodes are matched against nodes in said child node list according to said nodes' order in said list of pattern nodes and said child node list;

wherein said nodes are considered matched if said nodes have the same identification name;

wherein said identification name comprises at least a terminal name; and

wherein a report is generated indicating matched nodes.

73-76. (canceled).